LLaMat: Large Language Models for Materials Science Information Extraction

Vaibhav Mishra[*1], **Somaditya Singh**[*1], **Dhruv Ahlawat**[*1], **Mohd Zaki**[*2],
Hargun Singh Grover[3], **Biswajit Mishra**[4], **Santiago Miret**[5],
Mausam[1,3], **N. M. Anoop Krishnan**[2,3]
[1]Department of Computer Science and Engineering, [2]Department of Civil Engineering
[3]Yardi School of Artificial Intelligence, Indian Institute of Technology Delhi
[4]Cerebras Systems, Inc., [5]Intel labs
{mausam, krishnan}@iitd.ac.in

Abstract

Large language models have emerged as an important tool for information extraction and as scientific assistants in materials science and discovery. However, their performance is limited due to a lack of domain expertise. In this work, we propose LLaMat models, namely, LLaMat-2-7B and LLaMat-3-8B, which are obtained by continuously pre-training META's LLaMA-2-7B and LLaMA-3-8B models, respectively, on a large corpus of materials science text to improve their domain expertise. We also developed LLaMat-Chat models, the instruction fine-tuned variants of LLaMat models tailored through a dataset of one million instruction-output pairs, enabling interaction and information extraction abilities for the materials science domain. We show that LLaMat achieves state-of-the-art performance on several information extraction tasks from materials science text, where LLaMat-3-8B emerges as the best model. We also demonstrate the application of the developed model on structured information extraction capabilities of the developed chat models and compare their performance on 4 datasets ranging from named entity and relation extraction from text and understanding composition tables from materials science research papers.

1 Introduction

Knowledge about materials has been reported in the form of text, which includes books, research papers, patents, and technical reports, to name a few. It is humanly intractable for humans to go through a large amount of text and find answers to specific questions related to different materials science aspects[1, 2]. Dissemination of textual information in a natural language is an important aspect of democratising access to knowledge about materials science. However, developing a model capable of performing different types of tasks with high accuracy is a challenging task, which has been taken up by several researchers trying to address it by developing foundational models like large language models (LLMs) [3, 4].

LLMs have started revolutionizing both scientific and non-scientific domains. Due to their capability to perform a variety of tasks by understanding input in human language, they are also called foundational models. Recently, several researchers have attempted to develop and understand the capabilities of foundational models for chemistry ([5, 6]) and the medical domain([7]) or use general-purpose foundational models for domain-specific tasks either directly or after finetuning([8–12]). The benefits of domain adaptation of foundational models are well documented. Considering the wide variety of sub-domains in materials science, a foundational model will enable the researchers to get the answers to highly specialised questions.

AI4MAT workshop at 38th Conference on Neural Information Processing Systems (AI4MAT-NeurIPS 2024).

In response to the growing need for a foundational language model tailored to the domain of materials science, we propose LLaMat-2 and LLaMat-3 (Large Language Model for Materials Science). These models build upon the architecture of LLaMA-2-7B[13] and LLaMA-3-8B[14] respectively, undergoing further pretraining on a carefully curated corpus of high-quality materials science texts to enhance the models' domain-specific knowledge and performance on downstream tasks.

To provide LLaMat with conversational abilities, we introduce LLaMat-Chat models which are developed by instruction fine-tuning (IFT) on a dataset comprising $\approx$ one million instruction-output pairs. The IFT process equips the model with the capability to understand and generate responses based on given instructions, thus facilitating interactive and user-friendly applications. The chat models are proficient in performing classical natural language processing (NLP) tasks such as Named Entity Recognition, Abstract Classification, Relation Extraction, and Event Extraction for materials science datasets. In addition to these tasks, LLaMat-Chat models can also extract information in a structured way and understand complex data structures like tables from materials science research papers. Fig. 1 shows the pipeline of development of LLaMat and LLaMat-Chat models and their applications.

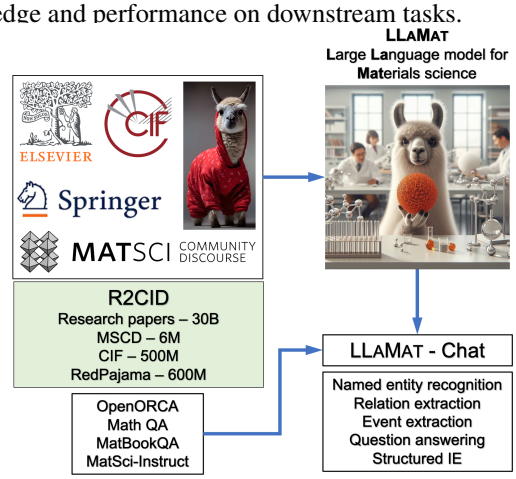

Figure 1: LLaMat and LLaMat-Chat development pipeline

2 R2CID - Pretrain Dataset

To obtain LLaMat models by continued pretraining of the LLaMA models, we consider the text from **R**esearch papers, a subset of **R**edpajama dataset, **Cif** (crystallographic information files) files **D**ataset. We call our training corpus the R2CID database. The details of each part are provided as follows.

Research Papers: We sourced research papers from around 500 Elsevier[15] journals and 300 Springer[16] journals to compile a comprehensive and high-quality dataset. The inclusion criteria required full-text availability in XML format for Elsevier papers and HTML format for Springer papers, ensuring compatibility with our data processing pipeline. The choice of Elsevier and Springer journals was influenced by the constraints of our institution's subscription contract, which provided access to a wide range of journals from these publishers. This contractual limitation shaped the scope of our dataset. The selected research papers' Digital Object Identifiers (DOIs) were retrieved using the CrossRef API[17]. After obtaining the DOIs, the full texts of the research papers were downloaded using the publisher specific APIs[18, 16]. These APIs facilitated access to the papers in the specified formats (XML for Elsevier and HTML for Springer), which were then incorporated into the R2CID corpus.

RedPajama Sample: The RedPajama dataset[19] was employed as the foundational corpus for the initial training phase of the LLaMA-2 model. We systematically extracted approximately 700 million tokens from this corpus to ensure a representative sample. The primary objective of incorporating this subset into R2CID is to address the issue of catastrophic forgetting, thereby preserving the model's comprehension and utility derived from its original, general-purpose training corpus. This ensures the model retains its foundational knowledge while effectively assimilating new information.

Crystallographic Information Files: While many text-based crystal representations exist [20], concrete material structures are often best obtained through diffraction studies and are reported as Crystallography Information Files. These are standardized text files used for storing and exchanging crystallographic data. These files contain unit cell parameters like the lengths of cell edges and angles between them. They also include symmetry information, such as the space group and symmetry operations, and atomic coordinates that specify the positions of atoms within the unit cell. To allow an increased understanding of CIF files, we considered a total of 470k CIF files and obtained their description in natural language using RoboCrystallographer[21]. The R2CID consists of these CIF

files and their descriptions from the Materials Project[22], Google GNoME[23], and AMSCSD database[24].

Merging the components to form R2CID: To enhance the effectiveness of model training and mitigate catastrophic forgetting, research papers were periodically interleaved with text from the RedPajama corpus. The periodic interleaving strategy was refined through a series of empirical evaluations. The selected interleaving period of 100 million research-related tokens with 2.3 million RedPajama tokens provided a balance that enhanced the model's ability to generalize and retain relevant information from both datasets. The CIF files were included in the posterior 10% of the corpus and interleaved with research papers.

3 Pretraining Methodology

LLaMAT-2 and LLaMAT-3 were initialised with weights of LLaMA-2-7B and LLaMA-3-8B, respectively, and then pretrained on R2CID for one epoch. The learning rate for both the models was initialised at 0, increased to 3×10^{-4} and 7×10^{-5} and then adhered to a cosine decay schedule to stop at 3×10^{-5} and 7×10^{-6} for LLaMAT-2 and LLaMAT-3, respectively. The pretraining of LLaMAT-2 was done using the Megatron-LLM library introduced by [25] and extended to LLaMA-2-7B by [7], which utilises 3D model parallelism for efficient training of LLMs. The pretraining was done on 16 A100 NVIDIA GPUs for approximately 9 days. We train LLaMAT-3 model on 2 $\times$ Cerebras CS-2 Wafer Scale Engine (WSE-2) in approximately 3 days. The CS-2 have $850,000$ core optimized for sparse linear algebra and 40 GB of on chip memory making it incredibly fast. It has a weight streaming based novel software stack that eliminates the need for model parallelism and enable linear scalability to hundred's of CS-2 without any code change [[26]].

4 Instruction Fine-Tune Methodology

LLaMAT-Chat models were initialized with the weights of the respective LLaMAT models. The instruction fine-tuning process was conducted in three distinct stages:

- **Stage 1:** LLaMAT-Chat was first fine-tuned on the OpenOrca dataset for one epoch. The objective of this stage was to enable the pretrained model to learn how to follow common English instructions.
- **Stage 2:** The model was further fine-tuned on a dataset of mathematical questions for three epochs. This stage aimed to enhance the mathematical reasoning capabilities of LLaMAT-Chat. Due to the relatively small size of this dataset, we observed a decrease in validation loss over the three epochs.
- **Stage 3:** In the final stage, LLaMAT-Chat was fine-tuned on a combined dataset constructed from MatSciInstruct, MatSciNLP, MatBookQA, and MaScQA (for one epoch).

The fine-tuning process was performed using the Megatron-LLM library. The learning rate for each stage was initialized at 2×10^{-6} and increased to 2×10^{-5} over the first 10% of the total iterations. Following this initial increase, the learning rate adhered to a cosine decay schedule. The same process was followed for obtaining chat models of LLaMAT-2 and LLaMAT-3.

5 Results

5.1 Downstream Tasks

To continuously evaluate the improved understanding of Materials Science principles gained by pretraining on **R2CID** and to measure any potential degradation in understanding conversational or informal English, we curated a dataset consisting of Materials Science and English Comprehension tasks. Table 1 shows the list of different tasks, datasets, and the number of samples in training and validation sets. The dataset has the following tasks: **sc**: sentence classification, **re**: relation extraction, **ner**: named entity extraction, **sar**: synthesis action retrieval, a type of classification task, **pc**: paragraph classification, **ee**: entity extraction, **sf**: slot filling, **qna**: question answering, and **mcq** : multiple choice question answering. The details of these tasks can be found in [27]. The samples from the training set were used to fine-tune the models before evaluation

Table 1: Details of downstream datasets

Task	Dataset	Train	Val
sc	sofc_sent	1893	1889
re	structured_re	1788	1786
ner	matscholar	1062	1061
ner	sc_comics	937	936
sar	synthesis_actions	565	569
re	sc_comics	376	373
pc	glass_non_glass	300	299
ee	sc_comics	287	288
ner	sofc_token	175	177
sf	sofc_token	175	179
qna	squad	1042	1042
mcq	hellaswag	981	980
mcq	boolqa	500	499

Table 2: Left:Performance on val set of downstream tasks, Right: Performance on test set of DISCOMAT dataset.

Model	Macro F1	Micro F1
LLaMA-2-7B	77.745	84.239
LLAMAT-2	82.26	87.85
LLAMAT-2-Chat	**84.66**	**89.51**
LLaMA-3-8B	80.827	87.636
LLAMAT-3	83.706	89.704
LLAMAT-3-Chat	**85.157**	**90.52**

Task	LLAMAT-2-7B	LLAMAT-3-8B
exact-match	513/720	395/589
comptable	0.81	0.835
regextable	0.857	0.789
glass id	0.695	0.59
composition labels	0.331	0.621
chemical labels	0.645	0.627

on the validation set to ensure that the models learned to follow the instructions. The performance of different models on these datasets is shown in Table 2. The Macro and Micro F1 scores were averaged over all the materials science tasks. It can be observed from Table 2 that domain-specific pretraining helped both LLaMA-2-7B and LLaMA-3-8B. However, the performance improvement is more for the former than the latter. Overall, the LLAMAT-3-Chat model emerges as the best on this dataset.

5.2 Structured information extraction

Extracting information in a structured manner allows ready conversion of extracted data to machine-readable form. Since the proposed LLAMAT models possess domain knowledge and information about structured data due to instruction fine-tuning, it is assumed these models shall perform reasonably well on such tasks. To this end, we take the chat models and pre-train them further on instruction-output pairs obtained by mixing four datasets: doping, general materials, metal-organic frameworks [8], and DISCOMAT [28]. The first three tasks are related to extracting named entities and their relations from the text, while the DISCOMAT dataset comprises tables from materials science research papers. To convert the first three datasets for instruction fine-tuning, we create six system prompts and use them as prefixes in a sentence comprising of the question, followed by delimiters and then the answer, which is in the form of a JSON schema as proposed in the literature [8](see Appendix 7.2.1). The original DiSCoMaT dataset was not meant for the instruction-finetuning of language models. Hence, we transform the existing annotations into JSON objects, which shall be generated by the language models when instructed. Considering the superior performance of LLAMAT-Chat models on downstream data, we evaluate only LLAMAT-Chat models on these datasets. Table 3 shows the F1 scores obtained using GPT-3, LLAMAT models on relation extraction tasks. Table 2 shows the performance of LLAMAT models on table understanding tasks (dataset detail in Appendix 7.2.2). It can be observed thatLLAMAT models exhibit reasonable performance for both tasks which can be improved further by training on more data or using methods like LoRA.

Table 3: Named entity recognition and relation extraction scores for three tasks in materials science using models with a JSON output schema

Task	Relation	GPT-3	LLAMAT-2-Chat	LLAMAT-3-Chat
Doping	host-dopant	0.726	0.396	**0.794**
General	formula - application	0.537	**0.644**	0.568
General	formula - description	0.354	0.208	**0.375**
General	formula - structure or phase	**0.482**	0.335	0.268
MOFs	name - applications	**0.573**	0.427	0.515
MOFs	name - guest species	0.616	**0.667**	0.491

6 Conclusion and future work

The results indicate that domain-specific continued pre-training improves performance on several tasks useful for materials discovery. The improvement in LLAMAT-2-Chat model over initial model is $\approx 7\%$ and 5% in macro and micro F1 scores as compared to LLAMAT-3-Chat model where these numbers are $\approx 4\%$ and 3%. For structured IE and table understanding tasks, the models show promising performance, exhibiting the wide range of capabilities of the proposed LLMs. Since the training corpus included information about different tasks related to materials discovery, like research papers, crystallography information files, information extraction tasks, and question-answering pairs, it will be interesting to evaluate the effect of each component of the corpus on the final performance of the model on specific tasks.

Acknowledgments

N. M. A. K. acknowledges the funding support received from BRNS YSRA (53/20/01/2021-BRNS), ISRO RESPOND as part of the STC at IIT Delhi, Google Research Scholar Award, Intel Labs, and Alexander von Humboldt Foundation. M. acknowledges grants by Google, IBM, Microsoft, Wipro, and a Jai Gupta Chair Fellowship. M. Z. acknowledges the funding received from the PMRF award by the Ministry of Education, Government of India. The authors thank Microsoft Accelerate Foundation Models Research (AFMR) for access to OpenAI models. The authors thank the High-Performance Computing (HPC) facility at IIT Delhi for computational and storage resources. This work was also supported by the Edinburgh International Data Facility (EIDF) and the Data-Driven Innovation Programme at the University of Edinburgh.

References

[1] Kausik Hira, Mohd Zaki, Dhruvil Sheth, NM Anoop Krishnan, et al. Reconstructing the materials tetrahedron: challenges in materials information extraction. *Digital Discovery*, 3(5): 1021–1037, 2024.

[2] Santiago Miret and NM Krishnan. Are llms ready for real-world materials discovery? *arXiv preprint arXiv:2402.05200*, 2024.

[3] Santiago Miret, NM Anoop Krishnan, Benjamin Sanchez-Lengeling, Marta Skreta, Vineeth Venugopal, and Jennifer N Wei. Perspective on ai for accelerated materials design at the ai4mat-2023 workshop at neurips 2023. *Digital Discovery*, 2024.

[4] Huan Zhang, Yu Song, Ziyu Hou, Santiago Miret, and Bang Liu. Honeycomb: A flexible llm-based agent system for materials science. *arXiv preprint arXiv:2409.00135*, 2024.

[5] Di Zhang, Wei Liu, Qian Tan, Jingdan Chen, Hang Yan, Yuliang Yan, Jiatong Li, Weiran Huang, Xiangyu Yue, Dongzhan Zhou, et al. Chemllm: A chemical large language model. *arXiv preprint arXiv:2402.06852*, 2024.

[6] Adrian Mirza, Nawaf Alampara, Sreekanth Kunchapu, Benedict Emoekabu, Aswanth Krishnan, Mara Wilhelmi, Macjonathan Okereke, Juliane Eberhardt, Amir Mohammad Elahi, Maximilian Greiner, et al. Are large language models superhuman chemists? *arXiv preprint arXiv:2404.01475*, 2024.

[7] Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, et al. Meditron-70b: Scaling medical pretraining for large language models. *arXiv preprint arXiv:2311.16079*, 2023.

[8] John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S Rosen, Gerbrand Ceder, Kristin A Persson, and Anubhav Jain. Structured information extraction from scientific text with large language models. *Nature Communications*, 15(1):1418, 2024.

[9] Maciej P Polak and Dane Morgan. Extracting accurate materials data from research papers with conversational language models and prompt engineering. *Nature Communications*, 15(1):1569, 2024.

[10] Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535, May 2024. ISSN 2522-5839. doi: 10.1038/s42256-024-00832-8.

[11] Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.

[12] Yu Song, Santiago Miret, Huan Zhang, and Bang Liu. HoneyBee: Progressive instruction finetuning of large language models for materials science. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5724–5739, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.380. URL https://aclanthology.org/2023.findings-emnlp.380.

[13] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[14] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[15] ScienceDirect.com | Science, health and medical journals, full text articles and books., . URL https://www.sciencedirect.com/.

[16] Springer Nature Developer Portal | APIs for Research Papers. URL https://dev.springernature.com/.

[17] Isaac Farley. Documentation. URL https://www.crossref.org/documentation/.

[18] Elsevier Developer Portal, . URL https://dev.elsevier.com/.

[19] togethercomputer/RedPajama-Data-1T · Datasets at Hugging Face, July 2024. URL https://huggingface.co/datasets/togethercomputer/RedPajama-Data-1T.

[20] Nawaf Alampara, Santiago Miret, and Kevin Maik Jablonka. Mattext: Do language models need more than text & scale for materials modeling? In *AI for Accelerated Materials Design-Vienna 2024*, 2024.

[21] Alex M. Ganose and Anubhav Jain. Robocrystallographer: automated crystal structure text descriptions and analysis. *MRS Communications*, 9(3):874–881, 2019. doi: 10.1557/mrc.2019.94.

[22] Anubhav Jain, Joseph Montoya, Shyam Dwaraknath, Nils ER Zimmermann, John Dagdelen, Matthew Horton, Patrick Huck, Donny Winston, Shreyas Cholia, Shyue Ping Ong, et al. The materials project: Accelerating materials design through theory-driven data and tools. *Handbook of Materials Modeling: Methods: Theory and Modeling*, pages 1751–1784, 2020.

[23] Amil Merchant, Simon Batzner, Samuel S Schoenholz, Muratahan Aykol, Gowoon Cheon, and Ekin Dogus Cubuk. Scaling deep learning for materials discovery. *Nature*, 624(7990):80–85, 2023.

[24] American Mineralogist Crystal Structure Database. URL https://rruff.geo.arizona.edu/AMS/amcsd.php.

[25] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

[26] Andrew Feldman. Linear Scaling Made Possible with Weight Streaming, September 2022.

[27] Yu Song, Santiago Miret, and Bang Liu. MatSci-NLP: Evaluating scientific language models on materials science language tasks using text-to-schema modeling. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3621–3639, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.201. URL https://aclanthology.org/2023.acl-long.201.

[28] Tanishq Gupta, Mohd Zaki, Devanshi Khatsuriya, Kausik Hira, N M Anoop Krishnan, and Mausam . DiSCoMaT: Distantly supervised composition extraction from tables in materials science articles. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13465–13483, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.753. URL https://aclanthology.org/2023.acl-long.753.

[29] Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*, 2023.

[30] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[31] Mohd Zaki, NM Anoop Krishnan, et al. Mascqa: investigating materials science knowledge of large language models. *Digital Discovery*, 3(2):313–327, 2024.

7 Appendix

7.1 Instruction Fine-Tune Dataset

We use various openly available instruction fine-tuning datasets related to Material science and general English question answering. We also construct a dataset for free-form question answering for material science questions by prompting GPT4 with a context and asking it to generate questions. We call this dataset MatBookQA (Material Science Book-based Question Answering dataset). We also introduce another question-answering dataset based on questions asked in the GATE examination in India, which is taken by undergraduate students to apply for admissions in Masters and PhD programs in premier institutes in India and some foreign institutions of repute. The details of each dataset are provided as follows.

7.1.1 OpenOrca

This dataset comprises $800,000$ high-quality and diverse textual instructions. A model fine-tuned on this dataset may demonstrate enhanced performance in comprehending technical jargon, responding to complex queries, and producing coherent and contextually appropriate text across various domains. Previous research, as detailed in [29], has demonstrated that large language models (LLMs) fine-tuned on this dataset outperform other models on a range of benchmarks.

7.1.2 Math

To induce the ability of mathematical problem-solving in our model, we train our model on the MATH dataset introduced by [30]. It consists of 7500 instructions aimed at complex mathematical reasoning.

7.1.3 MatSci

We utilize openly available instruction fine-tuning datasets for material science, complemented by a curated dataset generated through GPT-4(gpt-4-0613). By prompting GPT-4 with open-source material science textbooks, we elicit contextually complete questions covering various subdomains of material science. This diverse prompting ensures comprehensive coverage of the field.

We incorporate MatSciInstruct, as introduced in [12]. MatSciInstruct generates specialized instruction data through a two-step framework—Generation and Verification. In the Generation step, an instructor model creates domain-specific instruction data focused on materials science. The Verification step involves a separate verifier model for cross-verifying the instruction data for accuracy and relevance. Additionally, we employ the MatSciNLP training dataset and augment it with our MatBookQA dataset, as discussed below.

7.1.4 MatBookQA

We use an open-source book on Material Science and prompt GPT4 with one chapter at a time. We ask it to generate both short and long question-answer pairs for each chapter. We first curate a list of ten prompts each (see Appendix) to obtain short and long descriptions. This resulted in 2069 question-answer pairs, of which 1887 are short and 182 are long.

7.1.5 MaScQA

This dataset consists of 1036 and 549 questions from the civil and chemical engineering exams, respectively. The questions in this dataset can be divided into four types based on their structure: multiple-choice questions, matching-type questions, numerical answer questions with multiple choices, and numerical answer-based questions with no options. More details about the question structure can be found in Zaki et al. [31] An earlier version of MaScQA reported by Zaki et al.[31] also comprises 650 questions from the same materials science-related questions from the GATE exam. These questions come from various subdomains of materials science, like atomic structure, thermodynamics, electrical and magnetic behaviour of materials, materials manufacturing, applications, processing, and testing. Both these datasets cover vast subdomains of materials science, therefore serving as a challenging benchmark for evaluating the performance of large language models.

7.2 Structured information extraction downstream datasets

7.2.1 Named entity recognition and relation extraction dataset

This dataset is taken from [8] where authors have provided the input and output pairs for extracting different entities like host and dopants for the doping dataset; formula, name, acronym, applications, description, and structure or phase for the IE dataset for general materials science; and name, formula, application, and guest species for IE from text related to metal-organic frameworks (MOFs). The relation is established by connecting the extracted named entities from a given sentence. Hence, the performance on named entity extraction tasks influences the performance of relation extraction tasks.

7.2.2 DISCOMAT

This dataset was originally not prepared as an IFT dataset[28]. Therefore, we identify the task from the original dataset and prepare a JSON schema which can be easily used for IFT. The ***exact-match*** metric means the generated JSON is exactly the same as that of the original JSON. Another task is whether the given table is a composition table, which is reflected by the ***comptable*** task. Similarly, ***regex*** task means the composition from the given table can be extracted using domain-specific regular expression parsers. The ***glass id*** identification task involves generating the index of a table row or column where glass id, i.e., the unique id of materials present in the given table, is mentioned. The ***composition*** task comprises predicting the index of the rows or columns of the given table where the constituents corresponding to each material can be found. Finally, in the ***chemical*** task, the model has to generate the index of the rows or columns where constituent chemicals of the materials are present inside the given table.

A sample prompt along with the definition of different keys of JSON schema for table tasks are listed below:

```
You are an expert in materials science and extracting data from
tables. You have the fill the following dictionary
for the given table. Each key is defined as follows:
'comp_table'- If the input table has material compositions
then return [1], else [0];
'regex_table'- If the input table has material compositions and
they can be extracted using a regular expression
parser, then return [1], else [0]
'composition_row_index'-The list containing the index of rows
which have complete information about material composition.
'chemical_col_index'-The list containing the index of columns
which report values of constituent chemicals of the material.
'composition_col_index'-The list containing the index of columns
which have complete information about material composition.
'chemical_row_index'-The list containing the index of rows which
report values of constituent chemicals of the material.
'gid_row_index'-The index of row having material identifier.
'gid_col_index'-The index of column having material identifier.
dictionary =
{'comp_table': [],
'regex_table': [],
'composition_row_index': [],
'composition_col_index': [],
'chemical_row_index': [],
'chemical_col_index': [],
'gid_row_index': [],
'gid_col_index': []}
NOTE:The output will be the dictionary with keys having
non-empty lists ONLY
```

7.3 MatSci-NLP

To benchmark and compare the performance of LLaMat-Chat against other state-of-the-art models within the materials science domain, we utilized the MatSci-NLP dataset, a comprehensive benchmark for materials science NLP tasks [27]. The evaluation was conducted in a zero-shot manner. The substantial improvement in performance indicates that our pretraining corpus effectively imparts knowledge of various materials science principles to LLaMat-Chat, while our fine-tuning process enhances its instruction-following capabilities. Table 4 shows the performance of various models on the MatSci-NLP. Performance numbers for other models have been adapted from [12] while ensuring identical experimental settings. The scores are Macro-F1(Top) and Micro-F1(Bottom). It should be noted that the performance of all models except LLaMA-3-8B and LLaMat-Chat models is taken from [12]. It can be observed that materials domain specific pretraining has improved the performance of both LLaMat-2-Chat and LLaMat-3-Chat models, however, the performance of the former is slightly better than the latter. This may be explained by the similar choice of batch sizes, and learning rate parameters for both the models.

Table 4: Zero-shot performance of LLMs based on MatSci-NLP. The numbers in bold represent best scores, and the underlined numbers are the second best.

Model	Named Entity Recognition	Relation Extraction	Event Argument Extraction	Paragraph Classification	Synthesis Action Retrieval	Sentence Classification	Slot Filling	Overall (All Tasks)
Zero-Shot LLM Performance								
LLaMA-7b	0.042 0.064	0.094 0.013	0.160 0.042	0.279 0.218	0.052 0.013	0.096 0.087	0.142 0.010	0.208 0.064
LLaMA-13b	0.057 0.066	0.109 0.016	0.042 0.054	0.233 0.189	0.039 0.009	0.079 0.074	0.138 0.008	0.1 0.059
Alpaca-7b	0.031 0.018	0.053 0.037	0.029 0.009	0.375 0.294	0.179 0.129	0.180 0.180	0.139 0.039	0.141 0.101
Alpaca-13b	0.053 0.046	0.016 0.035	0.111 0.072	0.310 0.237	0.442 0.278	0.375 0.334	0.110 0.015	0.202 0.145
Chat-GPT	0.063 0.052	0.232 0.145	0.204 0.203	0.433 0.450	0.300 0.183	0.320 0.318	0.368 0.280	0.274 0.233
Claude	0.063 0.048	0.232 0.143	0.195 0.169	0.442 0.467	0.280 0.177	0.329 0.326	0.393 0.305	0.276 0.234
GPT-4	0.189 0.121	0.445 0.432	0.453 0.353	0.679 0.522	0.743 0.677	0.788 0.689	0.502 0.483	0.543 0.468
LLaMa-3-8B	0.591 0.675	0.816 0.852	0.676 0.787	0.631 0.832	0.891 0.891	0.924 0.951	0.727 **0.899**	0.751 0.841
LLaMat-2-Chat	**0.827** **0.898**	**0.968** **0.952**	0.633 **0.836**	**0.843** **0.871**	**0.938** **0.962**	0.773 0.917	**0.744** 0.839	**0.813** **0.894**
LLaMat-3-Chat	0.772 0.835	<u>0.861</u> <u>0.884</u>	**0.724** 0.824	<u>0.651</u> <u>0.856</u>	0.901 <u>0.902</u>	<u>0.941</u> <u>0.960</u>	0.712 <u>0.892</u>	0.795 0.879